

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): Sheng Ye et al.
Docket No.: CN920020009US1
Serial No.: 10/689,186
Filing Date: October 20, 2003
Group: 2176
Examiner: Nathan Hillery

Title: Method and Apparatus for Locating and Transforming Data

APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants (hereinafter referred to as “Appellants”) hereby appeal the final rejection, dated June 13, 2008, of claims 23-42 of the above-identified application.

REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corporation. The assignee, International Business Machines Corporation, is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no known related appeals or interferences.

STATUS OF CLAIMS

The present application was filed on October 20, 2003, and claims priority to Chinese Patent Application No. 02147150.9, filed on October 24, 2003.

The present application was filed with claims 1-19. In a response dated May 16, 2006, claims 1-11 were withdrawn. In a response dated November 22, 2006, claims 12-15 and 17-19 were canceled, and claims 20-22 were added. In a response dated March 25, 2008, all claims, including withdrawn claims 1-11 and pending claims 16 and 20-22, were canceled and replaced with new claims 23-42. Pursuant to 37 C.F.R. §1.145, claims 23-42 recite only subject matter corresponding to elected claims 12-19, rather than non-elected claims 1-11.

Claims 23-42 remain pending, including independent claims 23, 30 and 37.

Claims 23-42 stand finally rejected under 35 U.S.C. §112, first paragraph, and §102(b). Claims 23-42 are appealed.

STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 23 is directed to a method for generating at least a first output file from at least one input file, comprising the steps of defining at least one data unit of the at least one input file; determining one or more locations within the at least one input file of the at least one data unit; and transferring the at least one data unit from the one or more locations within the at least one input file to the one or more locations within the first output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the first output file. Each location comprises a horizontal position, the horizontal position comprising at least one of an uppermost position of the data unit or a lowermost position of the data unit, and a vertical position, the vertical position comprising at least one of the leftmost position of the data unit or the rightmost position of the data unit. Each data unit is defined based on at least one of: at least one string, at least one absolute position of the data unit within the input file, at least one relative position of the

data unit to a start or end of at least one of a row or column of the input file, and at least one relative position of the data unit to another data unit.

An exemplary method for generating at least a first output file (e.g., objective data files 120₁-120_N in FIG. 5) from at least one input file (e.g., original data files 110₁-110_N in FIG. 5) comprises the steps of defining at least one data unit of the at least one input file (e.g., step 501 in FIG. 6); determining one or more locations within the at least one input file of the at least one data unit (e.g., step 503 in FIG. 6); and transferring the at least one data unit from the one or more locations within the at least one input file to the one or more locations within the first output file (e.g., steps 504 and 505 in FIG. 6) specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the first output file (e.g., step 502 in FIG. 6). Each location comprises a horizontal position, the horizontal position comprising at least one of an uppermost position of the data unit or a lowermost position of the data unit, and a vertical position, the vertical position comprising at least one of the leftmost position of the data unit or the rightmost position of the data unit (see the specification at, for example, page 17, lines 10-14). Each data unit is defined based on at least one of: at least one string (see the specification at, for example, page 15, lines 6-7), at least one absolute position of the data unit within the input file (see the specification at, for example, page 18, lines 11-12), at least one relative position of the data unit to a start or end of at least one of a row or column of the input file (see the specification at, for example, page 18, lines 7-11), and at least one relative position of the data unit to another data unit (see the specification at, for example, page 18, lines 14-15).

Independent claim 30 is directed to an apparatus for generating at least a first output file from at least one input file; the apparatus comprises a memory, operative to store at least a portion of at least one of the at least first output file and the at least one input file, and at least one processor. The at least one processor is operative to perform the operations of defining at least one data unit of the at least one input file; determining one or more locations within the at least one input file of the at least one data unit; and transferring the at least one data unit from the one or more locations within the at least one input file to the one or more locations within the first output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the first output

file. Each location comprises a horizontal position, the horizontal position comprising at least one of an uppermost position of the data unit or a lowermost position of the data unit, and a vertical position, the vertical position comprising at least one of the leftmost position of the data unit or the rightmost position of the data unit. Each data unit is defined based on at least one of: at least one string, at least one absolute position of the data unit within the input file, at least one relative position of the data unit to a start or end of at least one of a row or column of the input file, and at least one relative position of the data unit to another data unit.

An exemplary apparatus for generating at least a first output file (e.g., objective data files 120₁-120_N in FIG. 5) from at least one input file (e.g., original data files 110₁-110_N in FIG. 5) comprises a memory, operative to store at least a portion of at least one of the at least first output file and the at least one input file, and at least one processor, as discussed in the specification at, for example, page 32, line 14, to page 33, line 3, and page 36, lines 5-6. The at least one processor is operative to perform the operations of defining at least one data unit of the at least one input file (e.g., step 501 in FIG. 6); determining one or more locations within the at least one input file of the at least one data unit (e.g., step 503 in FIG. 6); and transferring the at least one data unit from the one or more locations within the at least one input file to the one or more locations within the first output file (e.g., steps 504 and 505 in FIG. 6) specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the first output file (e.g., step 502 in FIG. 6). Each location comprises a horizontal position, the horizontal position comprising at least one of an uppermost position of the data unit or a lowermost position of the data unit, and a vertical position, the vertical position comprising at least one of the leftmost position of the data unit or the rightmost position of the data unit (see the specification at, for example, page 17, lines 10-14). Each data unit is defined based on at least one of: at least one string (see the specification at, for example, page 15, lines 6-7), at least one absolute position of the data unit within the input file (see the specification at, for example, page 18, lines 11-12), at least one relative position of the data unit to a start or end of at least one of a row or column of the input file (see the specification at, for example, page 18, lines 7-11), and at least one relative position of the data unit to another data unit (see the specification at, for example, page 18, lines 14-15).

Independent claim 37 is directed to a computer program product for generating at least a first output file from at least one input file. The computer program product comprising a computer usable medium having computer usable program code embodied therewith. The computer usable program code comprises computer usable program code configured to perform the operations of defining at least one data unit of the at least one input file; determining one or more locations within the at least one input file of the at least one data unit; and transferring the at least one data unit from the one or more locations within the at least one input file to the one or more locations within the first output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the first output file. Each location comprises a horizontal position, the horizontal position comprising at least one of an uppermost position of the data unit or a lowermost position of the data unit, and a vertical position, the vertical position comprising at least one of the leftmost position of the data unit or the rightmost position of the data unit. Each data unit is defined based on at least one of: at least one string, at least one absolute position of the data unit within the input file, at least one relative position of the data unit to a start or end of at least one of a row or column of the input file, and at least one relative position of the data unit to another data unit.

An exemplary computer program product for generating at least a first output file (e.g., objective data files 120₁-120_N in FIG. 5) from at least one input file (e.g., original data files 110₁-110_N in FIG. 5) comprises a computer usable medium having computer usable program code embodied therewith, as discussed in the specification at, for example, page 32, line 14, to page 33, line 3, and page 36, lines 5-6. The computer usable program code comprises computer usable program code configured to perform the operations of defining at least one data unit of the at least one input file (e.g., step 501 in FIG. 6); determining one or more locations within the at least one input file of the at least one data unit (e.g., step 503 in FIG. 6); and transferring the at least one data unit from the one or more locations within the at least one input file to the one or more locations within the first output file (e.g., steps 504 and 505 in FIG. 6) specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the first output file (e.g., step 502 in FIG. 6). Each location comprises a horizontal position, the horizontal position comprising at least one of an uppermost position of the data unit or a lowermost position of the data

unit, and a vertical position, the vertical position comprising at least one of the leftmost position of the data unit or the rightmost position of the data unit (see the specification at, for example, page 17, lines 10-14). Each data unit is defined based on at least one of: at least one string (see the specification at, for example, page 15, lines 6-7), at least one absolute position of the data unit within the input file (see the specification at, for example, page 18, lines 11-12), at least one relative position of the data unit to a start or end of at least one of a row or column of the input file (see the specification at, for example, page 18, lines 7-11), and at least one relative position of the data unit to another data unit (see the specification at, for example, page 18, lines 14-15).

Illustrative embodiments of the present invention provide numerous advantages over conventional techniques. For example, illustrative embodiments provide increased flexibility in extracting, transforming and locating data within files. See the specification at, for example, page 11, line 21, to page 12, line 13; page 33, lines 5-8; and page 35, lines 12-21.

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 23-42 are rejected under 35 U.S.C. §112, first paragraph, as failing to comply with the written description and enablement requirements.
2. Claims 23-42 are rejected under 35 U.S.C. §102(b) as being anticipated by Michael Barr, *Programming Embedded Systems in C and C++* (hereinafter “Barr”).

ARGUMENT

1. Rejection of claims 23-42 under 35 U.S.C. §112, first paragraph

With regard to the rejection under §112, first paragraph, for failing to comply with the written description requirement, Appellants note that the Examiner has the initial burden of presenting evidence or reasoning to explain why persons skilled in the art would not recognize in the original disclosure a description of the invention defined by the claims. See *In re Wertheim*, 541 F.2d 257, 263, 191 USPQ 90, 97 (CCPA 1976) (“[T]he PTO has the initial burden of presenting evidence or reasons why persons skilled in the art would not recognize in the disclosure a description of the invention defined by the claims.”). See also MPEP 2163 (where the “applicant amends the claims

and points out where and/or how the originally filed disclosure supports the amendment(s), and the examiner finds that the disclosure does not reasonably convey that the inventor had possession of the subject matter of the amendment at the time of the filing of the application, the examiner has the initial burden of presenting evidence or reasoning to explain why persons skilled in the art would not recognize in the disclosure a description of the invention defined by the claims.”)

Moreover, MPEP 2163 specifically states that “there is no *in haec verba* requirement [for] newly added claim limitations;” see also MPEP 2163.02 (“The subject matter of the claim need not be described literally (i.e., using the same terms or *in haec verba*) in order for the disclosure to satisfy the description requirement.”)

With regard to the rejection under §112, first paragraph, based on the enablement requirement, Appellants initially note that the Examiner has the initial burden to establish a reasonable basis to question the enablement provided for the claimed invention. *In re Wright*, 999 F.2d 1557, 1562, 27 USPQ2d 1510, 1513 (Fed. Cir. 1993) (examiner must provide a reasonable explanation as to why the scope of protection provided by a claim is not adequately enabled by the disclosure). A specification disclosure which contains a teaching of the manner and process of making and using an invention in terms which correspond in scope to those used in describing and defining the subject matter sought to be patented must be taken as being in compliance with the enablement requirement of §112, first paragraph, unless there is a reason to doubt the objective truth of the statements contained therein which must be relied on for enabling support. See MPEP 2164.04.

It should be further noted, even if one accepted the Examiner’s allegation that certain limitations of claim 23 lack descriptive support in the disclosure as originally filed, this does not necessarily mean that the limitation is also not enabled. Rather, the statement of a new limitation in and of itself may enable one skilled in the art to make and use the claim containing that limitation even though that limitation may not be described in the original disclosure. See MPEP 2164.

The Examiner contends that certain limitations recited in independent claim 23 identified in the final Office Action as limitations “A,” “B,” and “C,” are neither described nor enabled by the originally-filed specification. Appellants respectfully disagree.

Limitation “A” is directed to transferring the at least one data unit from the one or more locations within the at least one input file to the one or more locations within the first output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the first output file. This limitation is described by the specification at, for example, page 34, lines 4-7, with reference to FIG. 6:

At step S505, the extracted data is transformed into the data in the objective data files, in which the data units extracted at step S504 are transformed into the data in the objective data files, based on the correspondence between the one or more data units to be located and the specific formats of the objective data files.

In the Advisory Action dated October 16, 2008 (hereinafter “Advisory Action”), the Examiner argues that “limitation ‘A’ recites the transference of data units specified by a mapping. The accompanying support cited by the applicant discloses transforming data. . . . Thus, the specification does not enable” this limitation.

Appellants respectfully submit that limitation “A” is clearly supported by the specification at, for example, page 31, lines 11-19, and page 32, lines 8-12 (emphasis added):

The format mapping unit 402 builds up the correspondence between the data units located by the data locating device 401 and the format of the objective data file.

When the location descriptions of the data units in a source data file needed to be located are generated, each of the data units determined to be located is mapped to the corresponding format (corresponding area) of the objective data file, according to the format feature of the objective data file, so as to transform the data in the original data file into the data in the objective data file. Then the corresponding relationship between the data units to be located and the format of the objective data file need to be built up. . . .

The data transforming unit 405 transforms the data units extracted by the data extracting unit 404 into the data in the objective data file having a specific format or outputs the data into the corresponding areas in the objective data file, based on the correspondence relationship between the data units to be located and the specific format of the objective data file built up by the format mapping unit 402.

See also the specification at, for example, page 4, lines 3-4; page 7, lines 12-14; page 9, lines 1-4; page 11, lines 16-19; and page 34, lines 17-18.

Limitation “B” specifies that each location comprises a horizontal position, the horizontal position comprising at least one of an uppermost position of the data unit or a lowermost position of the data unit, and a vertical position, the vertical position comprising at least one of the leftmost position of the data unit or the rightmost position of the data unit. This limitation is supported by the specification at, for example, page 17, lines 10-14:

Four types of location elements are used to determine the position of a data unit, according to the invention, that is, the “top (Top)” representing the uppermost position of the data unit; the “bottom (Bottom)” representing the lowest position of the data unit; the “left (Left)” representing the most left position of the data unit, and the “right (Right)” representing the most right position of the data unit.

Further support for this limitation may be found in the specification at page 19, line 14, to page 20, line 20, as well as FIGS. 2 and 4.

In the Advisory Action, the Examiner argues that “the accompanying support cited by applicant discloses that it takes four types of location elements to determine a single position of a data unit.” Thus, the Examiner apparently contends that the originally-filed specification requires that a data unit must be specified using an uppermost position of the data unit, a lowermost position of the data unit, the leftmost position of the data unit and the rightmost position of the data unit, and that any other arrangement is the product of “hindsight . . . that was not apparent or available at the time of the time of the originally filed specification.”

Appellants respectfully disagree. The specification at page 22, lines 10-13, states that the “description includes the type of the data unit, one or more location elements for locating the data unit. The location elements includes [sic] ‘Top’, ‘Bottom’, ‘Left’ and ‘Right’.” (emphasis added). Similar statements are also found in the specification at, for example, page 4, lines 19-20; page 6, lines 8-9; page 8, lines 5-6; and page 10, lines 16-17.

Indeed, the specification provides specific examples of data units described using a subset of the four enumerated location elements. See, for example, the specification at page 19, line 20, to page 20, line 5, and at page 26, line 6, to page 27, line 20, each of which includes a description of a data unit using only “Top,” “Left” and “Right” location elements and without a “Bottom” location element. More generally, see the specification at, for example, page 20, lines 12-13 (“It will be apparent for the persons in the art to flexibly employ the attributes to describe the location elements of a data unit.”).

Limitation “C” specifies that each data unit is defined based on at least one of: at least one string, at least one absolute position of the data unit within the input file, at least one relative position of the data unit to a start or end of at least one of a row or column of the input file, and at least one relative position of the data unit to another data unit.

See, for example, the specification at page 20, line 22, to page 21, line 4:

The method for locating a data unit according to the invention includes at least the following manners. The first is to locate a data unit by string matching, the second is to locate a data unit by the relative coordinates between the data units, the third is to locate a data unit by the absolute coordinates of the data unit. The developers may use any of them or any combination of them to define the attributes of the location elements of a data unit.

The recited at least one string is further supported by the specification at, for example, page 15, lines 6-7, with reference to FIG. 2 (“[T]he ‘Text’ represents the string capable of being located and matched. For example, the data unit A in FIG. 2 is defined as a ‘Text’ data unit.”) The recited at least one absolute position of the data unit within the input file is further supported by the specification at, for example, page 18, lines 11-12 (“The ‘Base’ may also be an original point in an absolute coordinate.”) The recited at least one relative position of the data unit to another data unit is further supported by the specification at, for example, page 18, lines 14-15 (“The data units B, C and D may be defined by the position relationship with the data unit A, and some basic location elements”)

The recited at least one relative position of the data unit to a start or end of at least one of a row or column of the input file is supported by the specification at, for example, page 18, lines 7-11:

Specifically, the value of the attribute “Base” may be the ID of a “data unit” having been located, such as the ID of the data unit having the type of “Text”, or “the start of a line (RB)”, “the end of a line (RE)”, “the start of a column (CB)” and “the end of a column (CE)”. “The start of a line (RB)”, “the end of a line (RE)”, “the start of a column (CB)” and “the end of a column (CE)” all refer to the line or the column the “Base” currently located.

In the Advisory Action, the Examiner writes that “limitation ‘C’ recites that each data unit is defined based on a number of options including a string. The cited portions of the specification, however, disclose that data units mainly consist of five types [including] text.”

Appellants respectfully submit that the specification describes the “Text” type data unit as being defined based on a string. See the specification at page 15, lines 6-7, with reference to FIG. 2 (“[T]he ‘Text’ represents the string capable of being located and matched. For example, the data unit A in FIG. 2 is defined as a ‘Text’ data unit.”). See also the specification at page 23, lines 5-6 (“At step S303, when the type is “Text”, the step proceeds to S306, where the locating process is performed by string matching.”)

More generally, the Examiner accuses the Appellants of “mak[ing] the biggest leap” in regard to limitation “C” by allegedly arguing that “because the specification discloses that the invention should not be limited to the [enumerated] five types data units and that any data units for data locating may flexibly be incorporated when needed; then, the applicant can claim any type of data unit with any data limitation.”

Appellants respectfully submit that the Examiner has grossly mischaracterized both the scope of limitation “C” and the Appellants’ arguments. Limitation “C” in fact restricts the scope of claim 23 by specifying that each data unit must be defined based on at least one of a series of four enumerated options, each of which is explicitly described and enabled by the originally-filed specification.

For example, the first, second and fourth “options” recited in limitation “C,” namely “at least one string,” “at least one absolute position of the data unit within the input file,” and “at least one

relative position of the data unit to another data unit,” correspond respectively to the first, third, and second “manners” enumerated in the specification at page 20, line 22, to page 21, line 4:

The first is to locate a data unit by string matching, the second is to locate a data unit by the relative coordinates between the data units, the third is to locate a data unit by the absolute coordinates of the data unit.

The third “option” recited in limitation “C,” namely, “at least one relative position of the data unit to a start or end of at least one of a row or column of the input file,” is described and enabled by the specification at, for example, page 18, lines 7-11:

Specifically, the value of the attribute “Base” may be the ID of a “data unit” having been located, such as the ID of the data unit having the type of “Text”, or “the start of a line (RB)”, “the end of a line (RE)”, “the start of a column (CB)” and “the end of a column (CE)”. “The start of a line (RB)”, “the end of a line (RE)”, “the start of a column (CB)” and “the end of a column (CE)” all refer to the line or the column the “Base” currently located.

In view of the above, claims 23-42 are believed to be both described by and enabled by the originally-filed specification in accordance with 35 U.S.C. §112, first paragraph.

2. Rejection of claims 23-42 under 35 U.S.C. §102(b)

Claims 23-26, 29-33, 36-39 and 42

Appellants initially note that the Federal Circuit has recently stated that “unless a reference discloses within the four corners of the document not only all of the limitations claimed but also all of the limitations arranged or combined in the same way as recited in the claim, it cannot be said to prove prior invention of the thing claimed and, thus, cannot anticipate under 35 U.S.C. §102.” *Net MoneyIN Inc. v. VeriSign Inc.*, 545 F.3d 1359, 1369, 88 USPQ2d 1751, 1760 (Fed. Cir. 2008)

Claim 23 includes a limitation, heretofore identified as limitation “B,” specifying that each location comprises a horizontal position and a vertical position. These locations refer to locations of a data unit within an input and/or output file. The Examiner argues that this limitation is disclosed by Barr at page 23, second paragraph, which describes “a symbol table somewhere in the object file

that contains the names and locations of all the variables and functions referenced within the source file.” However, the location recited in Barr does not comprise a horizontal position and a vertical position, but rather is a single numerical value defining a logical address in terms of a memory offset. See Barr at page 23, fifth paragraph (“[I]f foo is located at offset 14 of the output data section, its entry in the symbol table will now contain that address.”)

Claim 23 also includes a limitation, heretofore identified as Limitation “C,” which specifies that each data unit is defined based on at least one of: at least one string, at least one absolute position of the data unit within the input file, at least one relative position of the data unit to a start or end of at least one of a row or column of the input file, and at least one relative position of the data unit to another data unit.

The Examiner argues that this limitation is met by Barr at page 23, first paragraph, which specifies that “each of these sections contains one or more blocks of code or data that originated within the original source file.” Appellants respectfully submit that the relied-upon portion of Barr fails to teach that the specific arrangement recited in claim 23 wherein each data unit is defined based on a string and/or an absolute or relative position.

In the Advisory Action, the Examiner contends that “absent enablement and guidance from the specification, the office was forced to rely on the knowledge known to one skill in the art in rejecting the claim limitations” and therefore “the reference anticipates the claim limitations . . . in such [sic] much as could be understood in light of the rejections under 35 USC 112.”

As discussed above, Appellants respectfully assert that claims 23–42 are both described and enabled by the originally-filed specification and the §112 rejections are improper. Nonetheless, Appellants respectfully submit that it is well-established law that even limitations which are unsupported by the originally-filed specification are to be fully considered in determining issues of patentability over the prior art. As such, the §112 rejections, even if valid, should be inapposite to the §102 rejection, despite the Examiner’s apparent contention to the contrary.

See *Ex parte Grasselli*, 231 USPQ 393 (Bd. App. 1983) (“All of these limitations of the claims must be considered regardless of whether or not they were supported by the specification as filed.”); *In re Miller*, 441 F2d 689, 169 USPQ 597 (C.C.P.A. 1971) (“[T]he section 102 rejection in

this case is not unlike the section 103 rejection which we *reversed* in *In re Wilson*, 57 CCPA 1029, 424 F.2d 1382, 165 USPQ 494 (1970), and is unsupportable for the same reason: ‘All words in a claim must be considered in judging the patentability of that claim against the prior art’”)

Independent claims 30 and 37 include limitations similar to those of claim 23, and are therefore believed to be patentable over the prior art for reasons similar to those described above with reference to claim 23.

With regard to claims 24-26 and 29, which depend from claim 23, claims 31-33 and 36, which depend from claim 30, and claims 38, 39 and 42, which depend from claim 37, Appellants assert that these claims are also patentable over the prior art of record by virtue of their dependency from their respective base claims.

Claims 27, 34 and 40

Claims 27, 34 and 40 are patentable over the prior art of record at least by virtue of their dependency from their respective base claims. Moreover, these claims also define separately patentable subject matter.

Dependent claim 27 includes a limitation directed to generating a second output file from the at least one input file by transferring the at least one instance of the data unit from the one or more locations within the at least one input file to the one or more locations within the second output file specified by the mapping of the at least one data unit of the at least one input file to one or more locations within the first output file.

In rejecting claim 27, the Examiner relies upon a portion of Barr which states that “[i]f you’ll be using more than one compiler (i.e., you’ll be writing parts of your program in different source languages), you need to make sure that each is capable of producing in the same format.”

Appellants respectfully submit that the relied-upon portion of Barr is clearly directed to generating different object files from different input files written in different source languages, rather than generating a plurality of output files from the same at least one input file, as recited in claim 27. Accordingly, Barr fails to teach or even suggest the limitations of claim 27.

Dependent claims 34 and 40 include limitations similar to those discussed with reference to claim 27, and are therefore believed to be patentable for similar reasons.

Claims 28, 35 and 41

Claims 28, 35 and 41 are patentable over the prior art of record at least by virtue of their dependency from their respective base claims. Moreover, these claims also define separately patentable subject matter.

Dependent claim 28 includes a limitation directed to generating a second output file from the at least one input file by transferring the at least one instance of the data unit from the one or more locations within the at least one input file to one or more locations within the new output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the second output file.

In rejecting claim 28, the Examiner relies upon a portion of Barr which states that “[i]f you’ll be using more than one compiler (i.e., you’ll be writing parts of your program in different source languages), you need to make sure that each is capable of producing in the same format.”

Appellants respectfully submit that the relied-upon portion of Barr is clearly directed to generating different object files from different input files written in different source languages, rather than generating a plurality of output files from the same at least one input file, as recited in claim 28. Accordingly, Barr fails to teach or even suggest the limitations of claim 28.

Dependent claims 35 and 41 include limitations similar to those discussed with reference to claim 28, and are therefore believed to be patentable for similar reasons.

In view of the above, Appellants believe that claims 23-42 are in condition for allowance, and respectfully request reversal of the pending rejections.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'David E. Shifren', written over a horizontal line.

David E. Shifren
Attorney for Applicant(s)
Reg. No. 59,329
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-2641

Date: January 13, 2009

CLAIMS APPENDIX

1-22. (Canceled)

23. A method for generating at least a first output file from at least one input file, comprising the steps of:

defining at least one data unit of the at least one input file;

determining one or more locations within the at least one input file of the at least one data unit; and

transferring the at least one data unit from the one or more locations within the at least one input file to the one or more locations within the first output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the first output file;

wherein each location comprises a horizontal position, the horizontal position comprising at least one of an uppermost position of the data unit or a lowermost position of the data unit, and a vertical position, the vertical position comprising at least one of the leftmost position of the data unit or the rightmost position of the data unit; and

wherein each data unit is defined based on at least one of: at least one string, at least one absolute position of the data unit within the input file, at least one relative position of the data unit to a start or end of at least one of a row or column of the input file, and at least one relative position of the data unit to another data unit.

24. The method of claim 23, wherein the at least one string is within the data unit.

25. The method of claim 24, wherein the at least one string is adjacent to the data unit.

26. The method of claim 23, wherein the step of transferring the at least one data unit comprises transforming the at least one data unit from a first format to a second format.

27. The method of claim 23, further comprising a step of generating a second output file from the at least one input file by transferring the at least one instance of the data unit from the one or more locations within the at least one input file to the one or more locations within the second output file specified by the mapping of the at least one data unit of the at least one input file to one or more locations within the first output file.

28. The method of claim 23, further comprising a step of generating a second output file from the at least one input file by transferring the at least one instance of the data unit from the one or more locations within the at least one input file to one or more locations within the new output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the second output file.

29. The method of claim 23, further comprising the step of generating a second output file from at least one new file by determining one or more locations within the at least another new file

of the at least one data unit of the at least one input file and transferring the at least one data unit from the determined one or more locations within the at least one new file to the one or more locations within the output file specified one or more locations within the new output file specified by the mapping of the at least one data unit of the at least one input file to one or more locations within the first output file.

30. An apparatus for generating at least a first output file from at least one input file, the apparatus comprising:

a memory, operative to store at least a portion of at least one of the at least first output file and the at least one input file;

at least one processor, operative to perform the operations of:

defining at least one data unit of the at least one input file;

determining one or more locations within the at least one input file of the at least one data unit; and

transferring the at least one data unit from the one or more locations within the at least one input file to the one or more locations within the first output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the first output file;

wherein each location comprises a horizontal position, the horizontal position comprising at least one of an uppermost position of the data unit or a lowermost position of the data unit, and a vertical position, the vertical position comprising at least one of the leftmost position of the data unit or the rightmost position of the data unit; and

wherein each data unit is defined based on at least one of: at least one string, at least one absolute position of the data unit within the input file, at least one relative position of the data unit to a start or end of at least one of a row or column of the input file, and at least one relative position of the data unit to another data unit.

31. The apparatus of claim 30, wherein the at least one string is within the data unit.

32. The apparatus of claim 30, wherein the at least one string is adjacent to the data unit.

33. The apparatus of claim 30, wherein the operation of transferring the at least one data unit comprises transforming the at least one data unit from a first format to a second format.

34. The apparatus of claim 30, wherein the processor is further operative to perform the operation of generating a second output file from the at least one input file by transferring the at least one instance of the data unit from the one or more locations within the at least one input file to the one or more locations within the second output file specified by the mapping of the at least one data unit of the at least one input file to one or more locations within the first output file.

35. The apparatus of claim 30, wherein the processor is further operative to perform the operation of generating a second output file from the at least one input file by transferring the at least one instance of the data unit from the one or more locations within the at least one input file to one

or more locations within the new output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the second output file.

36. The apparatus of claim 30, wherein the processor is further operative to perform the operation of generating a second output file from at least one new file by determining one or more locations within the at least another new file of the at least one data unit of the at least one input file and transferring the at least one data unit from the determined one or more locations within the at least one new file to the one or more locations within the output file specified one or more locations within the new output file specified by the mapping of the at least one data unit of the at least one input file to one or more locations within the first output file.

37. A computer program product for generating at least a first output file from at least one input file, the computer program product comprising a computer usable medium having computer usable program code embodied therewith, the computer usable program code comprising computer usable program code configured to perform the operations of:

defining at least one data unit of the at least one input file;

determining one or more locations within the at least one input file of the at least one data unit; and

transferring the at least one data unit from the one or more locations within the at least one input file to the one or more locations within the first output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the first output file;

wherein each location comprises a horizontal position, the horizontal position comprising at least one of an uppermost position of the data unit or a lowermost position of the data unit, and a vertical position, the vertical position comprising at least one of the leftmost position of the data unit or the rightmost position of the data unit; and

wherein each data unit is defined based on at least one of: at least one string, at least one absolute position of the data unit within the input file, at least one relative position of the data unit to a start or end of at least one of a row or column of the input file, and at least one relative position of the data unit to another data unit.

38. The computer program product of claim 37, wherein the at least one string is within the data unit or wherein the at least one string is adjacent to the data unit.

39. The computer program product of claim 37, wherein the operation of transferring the at least one data unit comprises transforming the at least one data unit from a first format to a second format.

40. The computer program product of claim 37, the computer usable program code further comprising computer usable program code configured to perform the operation of generating a second output file from the at least one input file by transferring the at least one instance of the data unit from the one or more locations within the at least one input file to the one or more locations

within the second output file specified by the mapping of the at least one data unit of the at least one input file to one or more locations within the first output file.

41. The computer program product of claim 37, the computer usable program code further comprising computer usable program code configured to perform the operation of generating a second output file from the at least one input file by transferring the at least one instance of the data unit from the one or more locations within the at least one input file to one or more locations within the new output file specified by a mapping of the at least one data unit of the at least one input file to one or more locations within the second output file.

42. The computer program product of claim 37, the computer usable program code further comprising computer usable program code configured to perform the operation of generating a second output file from at least one new file by determining one or more locations within the at least another new file of the at least one data unit of the at least one input file and transferring the at least one data unit from the determined one or more locations within the at least one new file to the one or more locations within the output file specified one or more locations within the new output file specified by the mapping of the at least one data unit of the at least one input file to one or more locations within the first output file.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.